
A fast and flexible machine learning approach to data quality monitoring

Gaia Grosso

Dipartimento di Fisica e Astronomia
Università di Padova
INFN, Sez. di Padova
Padova, Italy
CERN, Experimental Physics Department
Geneva, Switzerland
gaia.grosso@cern.ch

Nicolò Lai

Dipartimento di Fisica e Astronomia
Università di Padova
Padova, Italy
nicolo.lai@studenti.unipd.it

Marco Letizia

MaLGa Center - DIBRIS
Università di Genova
INFN, Sez. di Genova
Genova, Italy
marco.letizia@edu.unige.it

Jacopo Pazzini

Dipartimento di Fisica e Astronomia
Università di Padova
INFN, Sez. di Padova
Padova, Italy
jacopo.pazzini@unipd.it

Marco Rando

MaLGa Center - DIBRIS
Università di Genova
Genova, Italy
marco.rando@edu.unige.it

Andrea Wulzer

Dipartimento di Fisica e Astronomia
Università di Padova
Padova, Italy
andrea.wulzer@cern.ch

Marco Zanetti

Dipartimento di Fisica e Astronomia
Università di Padova
INFN, Sez. di Padova
Padova, Italy
Marco.Zanetti@cern.ch

Abstract

We present a machine learning based approach for real-time monitoring of particle detectors. The proposed strategy evaluates the compatibility between incoming batches of experimental data and a reference sample representing the data behavior in normal conditions by implementing a likelihood-ratio hypothesis test. The core model is powered by recent large-scale implementations of kernel methods, nonparametric learning algorithms that can approximate any continuous function given enough data. The resulting algorithm is fast, efficient and agnostic about the type of potential anomaly in the data. We show the performance of the model on multivariate data from a drift tube chambers muon detector.

1 Introduction

Modern high-energy physics experiments consist of complex detectors where hundreds of millions of sensors are read out as frequently as every few nanoseconds. The electrical signals are amplified, processed and combined before the trigger selection and the final storage. At each of these steps some errors can occur and invalidate the whole process. Monitoring systems are deployed to assess the quality of the data and keep the flow under control throughout all the stages of the acquisition. Data quality monitoring (DQM) is a challenging task from a statistical point of view due to its high intrinsic dimensionality and the high level of human supervision required. The unforeseen events incoming during an experimental run can be several. Some of them can be anticipated and recognised while they occur, others cannot. The detection of well known dysfunctions could be nonetheless missed due to the huge amount of channels that should be simultaneously monitored. Developing flexible highly automatized techniques to supervise multiple variables at once is thus fundamental to reduce the risk of failures [1, 2, 3, 4]. This work proposes the use of a recent machine learning approach to compare collected data with a sample of reference events that depicts the correct detector readings. This reference sample can be, for instance, a set of measurements in a controlled scenario or simulated events. The basic idea is to perform a hypothesis test powered by a fast and flexible machine learning (ML) algorithm. In practice, we leverage the ability of binary classifiers to implicitly model the underlying data-generating distributions and estimate the likelihood ratio test statistics. This is then used to assess whether the hypothesis underlying the observed data (alternative hypothesis) agrees with the assumption of normal behavior (null hypothesis). If a set of measurements significantly deviates from the reference sample, the learned likelihood ratio can be used to characterize the anomalies in the feature space. To increase the model’s sensitivity, we take a large reference sample and reduce its statistical fluctuations. This work is inspired by the ML model introduced in Ref. [5] in the context of model-independent new physics searches in high-energy physics. In that work, the authors propose an algorithm based on kernel methods with clear advantages compared to similar neural network implementations in terms of training time while obtaining similar performance. This feature makes the kernel approach ideal for DQM, where fast training is essential for online analyses. The paper is organized as follows. In the next section we introduce the experimental setup and the algorithm input variables. These include a reference data set collected under standard conditions and smaller samples with anomalous controlled behaviors. The ML model and our core strategy are then described in Section 3, whereas an overview of the results is given in Section 4. Finally, the last section is devoted to conclusions and further developments.

2 Experimental setup and data samples

In this work we consider an experimental apparatus consisting of a set of drift tube (DT) chambers developed and installed at the Legnaro INFN National Laboratory (Fig.1, left). It represents a reduced-scale version of the ones installed in the CMS experiment at the LHC [6]. The fundamental element of the detector is a 70 cm long tube with a $4 \times 2.1 \text{ cm}^2$ cross section (Fig.1, bottom right). Within each tube, an electric field is produced by an anodic wire (+3.6 kV) laid in the center and two cathodic strips (-1.2 kV) at the sides; an additional pair of strips at an intermediate voltage (+1.8 kV) are placed above and below the wire to improve the field homogeneity. The tubes are filled with a Ar - CO₂ gas mixture (85% - 15%) whose molecules are ionized when charged particles traverse the sensitive volume. The electrons produced ionizing the gas drift with constant velocity along the field lines towards the wire, where they are collected. The arrival time is recorded by the front-end electronics, which amplify the signal and filter noise below a specific threshold (nominal at 100 mV). The particle’s position (with a left-right ambiguity) is linearly dependent on the drift time. The two parameters of the linear relation are the drift velocity, known after calibration, and the time pedestal, which can be provided by an external trigger or by means of a *mean-timer* technique [7].

A drift tube chamber comprises 64 tubes grouped into four layers of 16 each. The layers are staggered horizontally by half a cell. From the measurements collected from of a single chamber, it is thus possible to track the position and slope of the charged particles crossing the detector by performing a linear fit of the hits from the four layers (Fig.1, top right). The setup in Legnaro records muons from cosmic rays (at a sea-level rate of about 1 per minute per cm^2) and exploits plastic scintillators as an external trigger.¹

¹More precisely, the data are acquired continuously at 40 MHz, with the trigger information from the scintillators added to the data stream

Each detector channel provides an (absolute) time stamp once a signal is recorded. Since we are interested in muons crossing the detector, we group the hits corresponding to such events and we reconstruct the muon track out of the observed hits pattern.

The performances of the muons track reconstruction deteriorate as soon as the detector conditions are anomalous. The following list of relevant quantities can be exploited to assess the quality of the recorded data:

- Drift times (t_i): the four drift times associated with the muon track.
- Slope (ϕ): the angle formed by the muon track in the plane orthogonal to the anodic wires (measurement plane) with respect to the vertical axis.
- Number of hits (n_{hits}): the number of hits recorded in a time window of one second around the muon crossing time.

To test the performances of the DQM approach presented in this work, we reproduce two of the typical failures that occur during detector operations. In particular, we consider the eventuality of the cathodic strips voltage reduced to 25% of their nominal value (i.e., to -300 V) and that of the front-end thresholds reduced to 75, 50 and 25 mV, respectively 75%, 50% and 25% of the nominal value. A dedicated data collection campaign has been launched.² As a result, a set of data samples has been collected, one consisting of 3×10^5 events being the reference and the other four of approximately 10^4 events, each corresponding to one of the anomalous conditions mentioned above.

3 Methodology

Model design. To estimate the likelihood ratio, we train a binary classifier on a reference sample $S_0 = \{x_i\}_{i=1}^{n_0}$ and on a set of measurements $S_1 = \{x_i\}_{i=1}^{n_1}$ using a weighted logistic loss

$$\ell(y, f(x)) = \frac{n}{n_0}(1-y) \log(1 + e^{f_w(x)}) + \frac{n}{n_1}y \log(1 + e^{-f_w(x)}). \quad (1)$$

with $n = n_0 + n_1$ and where y is the class label that takes the zero value for S_0 and one for S_1 . The model parameters w are going to be learned from the data. The weights n/n_y are introduced to compensate for the data imbalance ($n_0 \gg n_1$) while keeping the statistical advantage of a large reference sample. The learned classifier $f_{\hat{w}}$ is then obtained by minimizing the average loss over the training set. By a standard computation, the learned function can be shown to approximate the likelihood ratio $f_{\hat{w}} \approx f^*(x) = \log \frac{p(x|1)}{p(x|0)}$ (see, for instance, appendix A of Ref. [5]). The log-likelihood ratio test statistics can then be easily evaluated on the measurements as

$$t_{\hat{w}}(S_1) = 2 \sum_{x \in S_1} f_{\hat{w}}(x). \quad (2)$$

In this work, we consider a learning algorithm based on kernel methods [8] of the form $f_w(x) = \sum_{i=1}^n w_i k_\sigma(x, x_i)$, where $k_\sigma(x, x_i) = \exp(-\|x - x_i\|^2/2\sigma^2)$ is the Gaussian kernel function and the width σ is a hyper-parameter. Kernel methods are universal in the large sample limit [9, 10] but they scale poorly with data size. For this reason, we consider here Falkon [11], a modern library which makes use of a number of algorithmic ideas, such as column sub-sampling, and takes full advantage of GPU hardware to extend the use of kernel methods to large scale scenarios with fast training and strong statistical guarantees.

Model selection and training. The training is performed by considering the empirical risk based on Eq. (1) with an added L2 regularization term [12] weighted by a hyper-parameter λ . While the machine learning model is at its core a binary classifier, we do not monitor typical classification metrics such as binary accuracy or the AUC, and our metric of interest is the likelihood ratio test statistics given in Eq. (2). Hyper-parameters are tuned following the prescription given in Ref. [5],

²<https://doi.org/10.5281/zenodo.7128223>

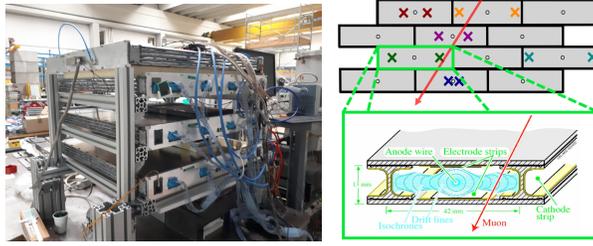


Figure 1: Left: the experimental apparatus at Legnaro Laboratory, with four drift-tube chambers vertically stacked. Right: a schematic view of the cell (bottom) and the hit pattern left by a charged particle crossing (top).

which includes a mix of heuristics, statistical optimality and efficiency requirements.

Anomaly detection. The analysis strategy for the detection of anomalies in the measurements is composed of three steps:

- The distribution of the test statistics under the null (reference) hypothesis is estimated by training the model multiple times ($\mathcal{O}(100)$) on the reference sample S_0 and on different data samples S_1 also following the reference distribution, e.g., anomaly free.
- A single training is performed on the reference S_0 and the actual measurements S_1 to estimate the test statistic $t_{\hat{w}}(S_1)$ of the measurements.
- The p -value corresponding to $t(S_1)$ is computed with respect to the test statistic distribution under the reference hypothesis, learned in the first step. If a statistically significant deviation from the reference data is found (for instance with respect to a threshold p -value p^*), the nature of the discrepancy can be analyzed in feature space by re-weighting the reference sample with the learned density ratio $\exp(f_{\hat{w}})$, as described in the following section.

If a database of observations characterizing known anomalous behaviors is available, different strategies can be employed to further characterize anomalous readings. We tested different ML approaches for this task, among which supervised binary classifiers, multi-class classifiers and parametrized regression models. These models are trained in a fully supervised fashion to separate reference data from specific types of anomalies. This step is not discussed in this manuscript and will be presented in details elsewhere.

4 Results

We present here the results obtained by applying this strategy to the monitoring of DT chambers, following the steps described in the previous section. Having real measurements at our disposal, this study represents a simplified but realistic scenario. As stated in 2, the input data is composed of six features (four drift times, the incoming angle, and the number of hits). For each training, we consider a reference of $n_0 = 10^4$ events, sampled from the totality of the available reference data, and a data sample of $n_1 = 10^3$ events (corresponding to approximately 10 minutes of data taking). We pre-process the data by dividing each feature by the standard deviation of the reference data. The model is trained on a single machine equipped with a NVIDIA Titan Xp GPU with 12 GB of VRAM. We first look at the full six-dimensional problem. In the left-side plot of Figure 2, we show the distribution of the likelihood ratio test statistics under the reference hypothesis (S_0 and S_1 both anomaly-free) and observe that it is well described by a χ^2 distribution (see Ref. [13, 14, 15]). In the right-side plot of Figure 2, we show the values of the test statistics with different types of anomalous measurements. The model is able to separate without false negatives the different types of anomalies in a coherent way. For instance, as the threshold gets lower, the data are recognized as departing more from the reference. The training time for a single classification step is approximately 1.5 seconds. We processed ten distinct data samples for each type of anomaly.

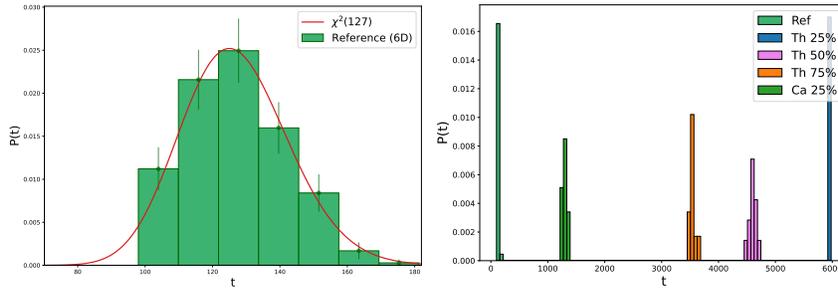


Figure 2: Distribution of the test statistics under the null hypothesis and for the anomalous data in 6D.

We now want to test the model when a given feature, which results to be particularly discriminant with respect to certain anomalies, is not provided as input. That is the case for n_{hits} with respect to the threshold anomalies, hence we remove it and perform new experiments in five dimensions. The distribution of the test statistics is reported in Figure 3. We see that the model is still able to successfully separate the various cases. In this case, the training time is reduced to approximately 0.5 seconds. To give an approximate but quantitative measure of the worst-case scenario, we estimate the

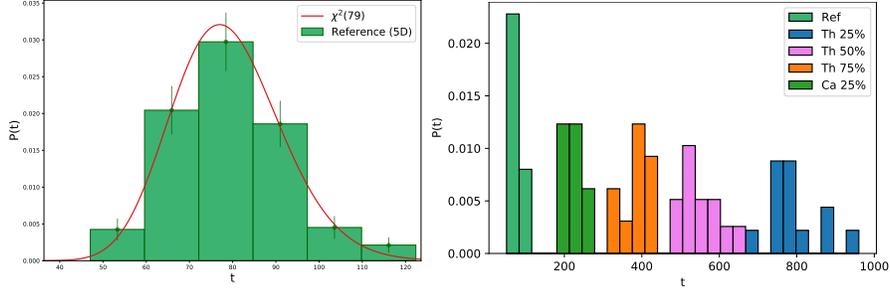


Figure 3: Distribution of the test statistics under the null hypothesis and for the anomalous data in 5D.

p-value associated with the anomalous data sample closest to the reference distribution in 5D. By using a χ^2 null distribution (left-side plot in Figure 3) we obtain $p \approx 10^{-11}$.

In Figure 4, we show instances of how to characterize potentially anomalous data by looking at the learned likelihood ratio. These plots are produced by re-weighting the features of the reference sample by $e^{f_{\tilde{w}}(x)}$ (evaluated on the reference training data), binning it and taking the ratio with the same binned reference sample (unweighted).³ This quantity is expected to be approximately one if no deviations are found in the measurements compared to the reference. We see that the model is able to successfully distinguish between statistical fluctuations (upper-left plot) and genuine anomalies (all the other plots).

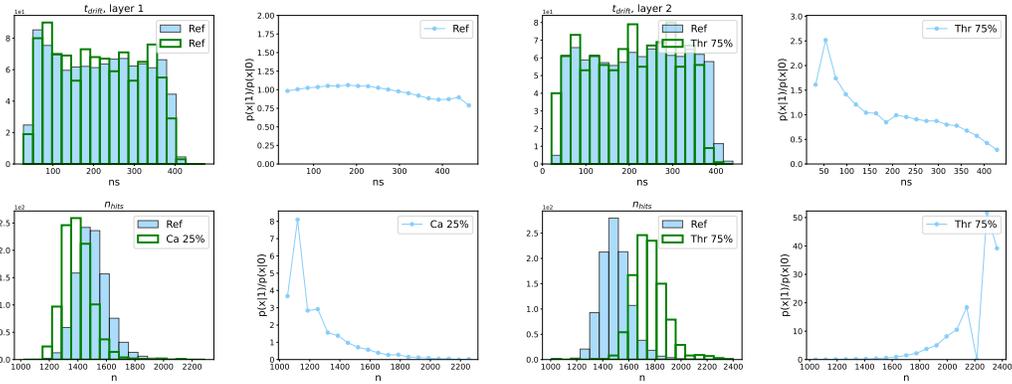


Figure 4: Examples of input data and respective learned likelihood ratios. Upper plots are produced without including n_{hits} among the input features.

5 Conclusions

In this work we propose a fast machine learning approach for data quality monitoring. The algorithm compares collected measurements with a reference dataset describing the standard detector readout performing a multidimensional likelihood-ratio hypothesis test. The significance of an eventual discrepancy in the data is quantified via a frequentist p-value. Some solutions for the location and characterization of the anomaly in the space of the input features are presented. Those could be relevant in order to design an automatic tool for the system control that allows to reset the apparatus to normal condition acting on the specific source of malfunctioning. The model is fast and can be used for quasi-online monitoring. The performances of the model can be tuned varying the size of the experimental data to be tested, the algorithm parameters and the computational resources. The experimental setup used to perform this study is simple but representative of the type of detectors installed in LHC experiments. A dedicated study tailoring more specific use cases will be performed in order to fit the requirements for the online deployment.

³We point out that the result of this operation is expected to be close but not identical to taking the ratio of the binned marginal distributions.

Acknowledgments and Disclosure of Funding

M.L and M.R. acknowledge the financial support of the European Research Council (grant SLING 819789). G.G. is supported by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (grant agreement no 772369). A.W. acknowledges support from the PRIN grant 2017FMJFMW.

References

- [1] Adrian Alan Pol, Gianluca Cerminara, Cecile Germain, and Maurizio Pierini. Data quality monitoring anomaly detection. In *Artificial Intelligence for High Energy Physics*, pages 115–149. World Scientific, 2022.
- [2] Adrian Alan Pol, Gianluca Cerminara, Cécile Germain, Maurizio Pierini, and Agrima Seth. Detector monitoring with artificial neural networks at the cms experiment at the cern large hadron collider. *Computing and Software for Big Science*, 3(1):1–13, 2019.
- [3] Virginia Azzolini, Broen van Besien, Dmitrijus Bugelskis, Tomas Hreus, Kaori Maeshima, Javier Fernandez Menendez, Antanas Norkus, James Fraser Patrick, Marco Rovere, and Marcel Andre Schneider. The Data Quality Monitoring software for the CMS experiment at the LHC: past, present and future. *EPJ Web Conf.*, 214:02003. 8 p, 2019.
- [4] M Adinolfi, F Archilli, W Baldini, A Baranov, D Derkach, A Panin, A Pearce, and A Ustyuzhanin. Lhcb data quality monitoring. In *Journal of Physics: Conference Series*, volume 898, page 092027. IOP Publishing, 2017.
- [5] Marco Letizia, Gianvito Losapio, Marco Rando, Gaia Grosso, Andrea Wulzer, Maurizio Pierini, Marco Zanetti, and Lorenzo Rosasco. Learning new physics efficiently with nonparametric methods. *Eur. Phys. J. C*, 82(10):879, 2022.
- [6] S. Chatrchyan et al. The CMS Experiment at the CERN LHC. *JINST*, 3:S08004, 2008.
- [7] Matteo Migliorini, Jacopo Pazzini, Andrea Triossi, Marco Zanetti, and Alberto Zucchetta. Muon trigger with fast Neural Networks on FPGA, a demonstrator. 5 2021. arXiv:2105.04428 [hep-ex].
- [8] T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer, 2009.
- [9] Charles A. Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research*, 7(95):2651–2667, 2006.
- [10] Andreas Christmann and Ingo Steinwart. *Support vector machines*. Springer, 2008.
- [11] Giacomo Meanti, Luigi Carratino, Lorenzo Rosasco, and Alessandro Rudi. Kernel methods through the roof: handling billions of points efficiently. *Advances in Neural Information Processing Systems*, 33:14410–14422, 2020. arXiv:2006.10350 [cs.LG].
- [12] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
- [13] S. S. Wilks. The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses. *Annals Math. Statist.*, 9(1):60–62, 1938.
- [14] Abraham Wald. Tests of statistical hypotheses concerning several parameters when the number of observations is large. *Transactions of the American Mathematical society*, 54(3):426–482, 1943.
- [15] Glen Cowan, Kyle Cranmer, Eilam Gross, and Ofer Vitells. Asymptotic formulae for likelihood-based tests of new physics. *Eur. Phys. J. C*, 71:1554, 2011. [Erratum: Eur.Phys.J.C 73, 2501 (2013)]. arXiv:1007.1727 [physics.data-an].

Checklist

1. For all authors...
 - (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
 - (b) Did you describe the limitations of your work? [Yes] **See the Conclusions.**
 - (c) Did you discuss any potential negative societal impacts of your work? [No] **We propose a machine learning model based on known statistical tools with the scope of performing hypothesis testing. Our applications are on high energy physics data only.**
 - (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
 - (a) Did you state the full set of assumptions of all theoretical results? [N/A]
 - (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
 - (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] **In the main text we present the full pipeline for the reproduction of the results. The machine learning model is built on top of existing public libraries. A link to the data is provided in Section 2**
 - (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] **In the section Model. We also reference other papers that inspired our work where these details are fully explained.**
 - (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] **Where relevant.**
 - (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] **Computational resources and training times are reported in Section 4**
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
 - (a) If your work uses existing assets, did you cite the creators? [Yes]
 - (b) Did you mention the license of the assets? [Yes]
 - (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
 - (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] **We used publicly available assets.**
 - (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] **We do not use data that contains personally identifiable information or offensive content.**
5. If you used crowdsourcing or conducted research with human subjects...
 - (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
 - (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
 - (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]